Appl Intell (2017) 47:1031–1043
DOI 10.1007/s10489-017-0943-y

A robust formulation for twin multiclass support vector machine

Julio López[1] · Sebastián Maldonado[2] · Miguel Carrasco[2]

Published online: 9 May 2017

Abstract Multiclass classification is an important task in pattern analysis since numerous algorithms have been devised to predict nominal variables with multiple levels accurately. In this paper, a novel support vector machine method for twin multiclass classification is presented. The main contribution is the use of second-order cone programming as a robust setting for twin multiclass classification, in which the training patterns are represented by ellipsoids instead of reduced convex hulls. A linear formulation is derived first, while the kernel-based method is also constructed for nonlinear classification. Experiments on benchmark multiclass datasets demonstrate the virtues in terms of predictive performance of our approach.

Keywords Support vector classification · Multiclass classification · Twin support vector machines · Second-order cone programming

✉ Sebastián Maldonado
smaldonado@uandes.cl

Julio López
julio.lopez@udp.cl

Miguel Carrasco
micarrasco@uandes.cl

[1] Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Ejército 441, Santiago, Chile

[2] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Monseñor Álvaro del Portillo 12455, Las Condes, Santiago, Chile

1 Introduction

Support Vector Machine (SVM) [35] is a very popular tool for multiclass learning, which has been used in various applications, such as computer vision applications [4], medical diagnosis [30], and financial analytics [18]. The most frequently used strategies are the construction of multiple binary SVM classifiers in one-vs.-one or a one-vs.-rest bipartite competition framework [5], and all-together approaches that aim at constructing all classifiers in a single optimization problem [6, 9, 39].

One of these numerous SVM multiclass approaches is twin support vector machine (Twin-KSVC) [42], which extends the ideas of twin SVM [17] for predicting nominal output variables with multiple levels. Twin SVM for binary classification constructs two nonparallel hyperplanes in such a way that each one is close to one of the two classes, and as far as possible from the other. The main advantage of this approach is the gain in efficiency since the original problem is split into two smaller sub-problems, leading to better running times. Additionally, the method may lead to better predictive results as well [17]. The Twin-KSVC method relies on a one-vs.-one-vs.-rest competition scheme, which means that each of the twin hyperplanes provides a three-label output, indicating whether the sample belongs to either class "+1", or class "-1", or to none of them (class "0"). Twin SVM has been successfully used in several applications, such as medical diagnosis [34], software fault prediction [1], and image processing [43].

Robust optimization has been used successfully for multiclass SVM classification [22, 46]. Here we distinguish two strategies. On the one hand, data may contain noise in the form of measurement errors, for example, and

some research papers report attempts to model this issue via second-order cone programming [15, 46]. For these approaches, a conic constraint is added for every noisy instance. On the other hand, a robust setting was proposed by Nath and Bhattacharyya [28], in which the traditional maximum margin approach for SVM is adapted by replacing the reduced convex hulls by ellipsoids. This strategy provides a framework that assures the right classification of each training pattern up to predefined rates, even for the worst possible data distribution.

In our work we extend the ideas described in [28] to the Twin-KSVC for both linear and kernel-based classification, providing a geometrical interpretation of the approach and an empirical comparison with the best-known SVM approaches for multiclass classification using benchmark datasets. Regarding novelty and previous works, the paper by Qi et al. [29] also reports combining robust optimization and twin SVM, but they follow the first strategy (modelling measurement errors), which is, again, a completely different approach compared to ours.

This work is structured as follows: Section 2 provides a brief description of developments for multiclass SVM classification. The proposed robust twin SVM method for multiclass classification is presented in Section 3. Section 4 describes our results using benchmark datasets. A summary of this paper can be found in Section 5, where we provide its main conclusions and address future developments.

2 Prior work in multiclass SVM classification

In this section, we describe the most commonly used multiclass SVM formulations (OVA-SVM, OVO-SVM, and MC-SVM), which were used as alternative methods in our experiments. Additionally, we present the following extensions to twin SVM: the OVA-TWSVM and Twin-KSVC methods, where the last method is closely related to our proposal.

2.1 One-versus-all approach

The One-vs.-All (OVA) SVM strategy aims at constructing K binary SVM classifiers independently, separating a given class from the others as a group [35]. Formally, for m training tuples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \Re^n$ is the i-th sample and $y_i \in \{1, 2, \ldots, K\}$ its respective class label, the k-th model built by of OVA-SVM has the following form:

$$\begin{array}{ll} \min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}_k} & \frac{1}{2}\|\mathbf{w}_k\|^2 + c\sum_{i=1}^{m}\xi_i^k \\ \text{s.t.} & \tilde{y}_i(\mathbf{w}_k^\top \mathbf{x}_i + b_k) \geq 1 - \xi_i^k, \quad i = 1, \ldots, m, \\ & \xi_i^k \geq 0, \quad i = 1, \ldots, m, \end{array} \tag{1}$$

where $\tilde{y}_i = 1$ means the sample i has label k ($y_i = k$), while $\tilde{y}_i = -1$ corresponds to the opposite case: object i belongs to a category different from k, and $c > 0$. The decision function for OVA-SVM is given by $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + b_k$, and a new sample $\mathbf{x}$ is assigned to the class with the highest value of $f_k(\mathbf{x})$ (i.e. $f_{k^*}(\mathbf{x}) = \max\{f_k(\mathbf{x}) : k = 1, \ldots, K\}$).

2.2 One-versus-one approach

Another well-known SVM variation is the One-versus-One (OVO) SVM [19], which constructs $K(K-1)/2$ binary SVM classifiers, one for each pair of categories. Given training points from classes k and l, OVO-SVM solves the following problem:

$$\begin{array}{ll} \min_{\mathbf{w}_{kl}, b_{kl}, \boldsymbol{\xi}^{kl}} & \frac{1}{2}\|\mathbf{w}_{kl}\|^2 + c\sum_r \xi_r^{kl} \\ \text{s.t.} & \mathbf{w}_{kl}^\top \mathbf{x}_r + b_{kl} \geq 1 - \xi_r^{kl}, \text{ if } y_r = k, \\ & -(\mathbf{w}_{kl}^\top \mathbf{x}_r + b_{kl}) \geq 1 - \xi_r^{kl}, \text{ if } y_r = l, \\ & \xi_r^{kl} \geq 0, \quad r = 1, \ldots, m_k + m_l, \end{array} \tag{2}$$

where m_k and m_l are the cardinality of the sets of training points of classes k and l, respectively. The decision function for a new instance $\mathbf{x}$ is given by $f_{kl}(\mathbf{x}) = \mathbf{w}_{kl}^\top \mathbf{x} + b_{kl}$. A max-wins voting strategy is used, in which each classification function assigns its respective data objects to one of the two categories, increasing by one the vote for the assigned class [13]. The category with most votes determines the classification of each new object.

2.3 "All-together" SVM approaches

Several multi-class (MC) SVM approaches that solve one single optimization problem have been proposed in the literature [2, 6, 9, 39]. For instance, in [39] the authors extends the ideas of OVA-SVM by constructing K binary classifiers simultaneously. The MC-SVM formulation follows:

$$\begin{array}{ll} \min_{\mathbf{w}_k, b_k, \xi_i^k} & \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2 + c\sum_{i=1}^{n}\sum_{k=1, k\neq y_i}^{K}\xi_i^k \\ \text{s.t.} & (\mathbf{w}_{y_i}^\top \mathbf{x}_i + b_{y_i}) - (\mathbf{w}_k^\top \mathbf{x}_i + b_k) \geq 2 - \xi_i^k, \\ & \xi_i^k \geq 0, \quad i = 1, \ldots, m, \ k \in \{1, \ldots, K\} \setminus y_i, \end{array} \tag{3}$$

where $\mathbf{w}_k, b_k$, for $k = 1, \ldots, K$, represent all the hyperplanes constructed by this approach. The decision rule is equivalent to that of the OVA-SVM formulation, in which a new sample $\mathbf{x}$ belongs to the class k^* iff $k^* = \text{argmax}_{k=1,\ldots,K}\{\mathbf{w}_k^\top \mathbf{x} + b_k\}$. Another "all-together" SVM formulation was the proposed in [9], in which K hyperplanes are also constructed but without a bias term.

2.4 One-versus-all twin support vector machine

This method constructs K nonparallel hyperplanes by solving K independent quadratic programming problems (QPPs), one for each class [41]. Formally, One-versus-All twin SVM (OVA-TWSVM) solves the following problem, for each class k ($k = 1, \ldots, K$):

$$\begin{array}{ll} \min_{\mathbf{w}_k, b_k, \boldsymbol{\xi}} & \frac{1}{2}\|A_k\mathbf{w}_k + b_k\mathbf{e}_k\|^2 + c\,\tilde{\mathbf{e}}_k^\top\boldsymbol{\xi} \\ \text{s.t.} & -(\tilde{A}_k\mathbf{w}_k + \tilde{\mathbf{e}}_k b_k) + \boldsymbol{\xi} \geq \tilde{\mathbf{e}}_k \\ & \boldsymbol{\xi} \geq 0, \end{array} \tag{4}$$

where $A_k \in \Re^{m_k \times n}$ and $\tilde{A}_k \in \Re^{m-m_k \times n}$ represent the data matrices for class k and for the remaining classes, respectively; c is a positive parameter; and $\mathbf{e}_k$ and $\tilde{\mathbf{e}}_k$ are vectors of ones of appropriate dimensions. For this approach, a new sample $\mathbf{x}$ belongs to the class k^* iff $k^* = \text{argmin}_{k=1,\ldots,K}\{\mathbf{w}_k^\top\mathbf{x} + b_k\}$. Another version of OVA-TWSVM was presented in [44].

2.5 Twin multi-class classification support vector machine

Twin multi-class classification support vector machine (Twin-KSVC) [42] is a new approach that extends the ideas of twin SVM [17] to a multiclass setting. This method evaluates all training points according to a "1-vs-1-vs-rest" framework with ternary output $\{-1, 0, +1\}$, similar to the K-SVCR method [2]. For each pair of classes, Twin-KSVC finds two non-parallel hyperplanes in $\Re^n$ of the form

$$\mathbf{w}_1^\top\mathbf{x} + b_1 = 0, \quad \mathbf{w}_2^\top\mathbf{x} + b_2 = 0, \tag{5}$$

in such a way that each hyperplane is close to one class, and as far as possible from the other. The remaining samples are mapped into a region between the two non-parallel hyperplanes.

Let us denote by $A \in \Re^{m_1 \times n}$ and $B \in \Re^{m_2 \times n}$ the two data matrices from the two target classes, which are labeled "+1" and "-1", respectively. We also denote by $C \in \Re^{m_3 \times n}$ a data matrix representing the remaining training samples, which are labeled "0". Formally, the linear Twin-SVC method solves the following two QPPs [42]:

$$\begin{array}{ll} \min_{\mathbf{w}_1, b_1, \boldsymbol{\xi}, \boldsymbol{\zeta}} & \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + c_1\mathbf{e}_2^\top\boldsymbol{\xi} + c_2\mathbf{e}_3^\top\boldsymbol{\zeta} \\ \text{s.t.} & -(B\mathbf{w}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}, \\ & -(C\mathbf{w}_1 + \mathbf{e}_3 b_1) \geq \mathbf{e}_3(1-\epsilon) - \boldsymbol{\zeta}, \\ & \boldsymbol{\xi},\ \boldsymbol{\zeta} \geq 0, \end{array} \tag{6}$$

and

$$\begin{array}{ll} \min_{\mathbf{w}_2, b_2, \boldsymbol{\xi}^*, \boldsymbol{\zeta}^*} & \frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + c_3\mathbf{e}_1^\top\boldsymbol{\xi}^* + c_4\mathbf{e}_3^\top\boldsymbol{\zeta}^* \\ \text{s.t.} & (A\mathbf{w}_2 + \mathbf{e}_1 b_2) \geq \mathbf{e}_1 - \boldsymbol{\xi}^*, \\ & (C\mathbf{w}_2 + \mathbf{e}_3 b_2) \geq \mathbf{e}_3(1-\epsilon) - \boldsymbol{\zeta}^*, \\ & \boldsymbol{\xi}^*,\ \boldsymbol{\zeta}^* \geq 0, \end{array} \tag{7}$$

where c_1, c_2, c_3, c_4, and ϵ are positive parameters; while $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are vectors of ones of appropriate dimensions. The set of parameters c_1, c_2, c_3 and c_4 determines the tradeoff between model fit and complexity [42].

For a new sample $\mathbf{x}$, Twin-KSVC determines its class label by the following decision function:

$$f(\mathbf{x}) = \begin{cases} +1\ , & \text{if } \mathbf{w}_1^\top\mathbf{x} + b_1 > -1 + \epsilon, \\ -1\ , & \text{if } \mathbf{w}_2^\top\mathbf{x} + b_2 < 1 - \epsilon, \\ 0\ , & \text{otherwise.} \end{cases} \tag{8}$$

A kernel-based classifier can be derived by considering the following non-linear surfaces:

$$\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_1 + b_1 = 0, \text{ and } \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_2 + b_2 = 0, \tag{9}$$

where $\mathbb{X} = [A^\top\ B^\top\ C^\top] \in \Re^{n \times m}$ represents the matrix of all training patterns, and $\mathcal{K}\colon \Re^n \times \Re^n \to \Re$ is a kernel function satisfying Mercer's condition [27].

A common choice is the *Gaussian kernel*, which is defined by $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2/2\sigma^2)$, where σ is a positive parameter that controls the width of the kernel [31].

For the above surfaces, the following quadratic problems can be constructed (kernel-based Twin-KSVC [42]):

$$\begin{array}{ll} \min_{\mathbf{u}_1, b_1, \boldsymbol{\xi}, \boldsymbol{\zeta}} & \frac{1}{2}\left\|\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_1 b_1\right\|^2 + c_1\mathbf{e}_2^\top\boldsymbol{\xi} + c_2\mathbf{e}_3^\top\boldsymbol{\zeta} \\ \text{s.t.} & -(\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_2 b_1) \geq \mathbf{e}_2 - \boldsymbol{\xi}, \\ & -(\mathcal{K}(C^\top, \mathbb{X})\mathbf{u}_1 + \mathbf{e}_3 b_1) \geq \mathbf{e}_3(1-\epsilon) - \boldsymbol{\zeta}, \\ & \boldsymbol{\xi}, \boldsymbol{\zeta} \geq 0, \end{array} \tag{10}$$

and

$$\begin{array}{ll} \min_{\mathbf{u}_2, b_2, \boldsymbol{\xi}^*, \boldsymbol{\zeta}^*} & \frac{1}{2}\left\|\mathcal{K}(B^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_2 b_2\right\|^2 + c_3\mathbf{e}_1^\top\boldsymbol{\xi}^* + c_4\mathbf{e}_3^\top\boldsymbol{\zeta}^* \\ \text{s.t.} & (\mathcal{K}(A^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_1 b_2) \geq \mathbf{e}_2 - \boldsymbol{\xi}^*, \\ & (\mathcal{K}(C^\top, \mathbb{X})\mathbf{u}_2 + \mathbf{e}_3 b_2) \geq \mathbf{e}_3(1-\epsilon) - \boldsymbol{\zeta}^*, \\ & \boldsymbol{\xi}^*, \boldsymbol{\zeta}^* \geq 0, \end{array} \tag{11}$$

where c_1, c_2, c_3, and c_4 are positive parameters. The corresponding decision function in the nonlinear Twin-KSVC case is

$$f(\mathbf{x}) = \begin{cases} +1\ , & \text{if } \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_1 + b_1 > -1 + \epsilon, \\ -1\ , & \text{if } \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{u}_2 + b_2 < 1 - \epsilon, \\ 0\ , & \text{otherwise.} \end{cases} \tag{12}$$

3 Twin-KSOCP, a robust twin multiclass SVM classifier

In this section, we introduce a novel multiclass approach based on second-order cones, extending the ideas of Twin-KSVC to robust optimization. The main idea is to construct two twin classifiers for each pair of classes according to the "1-vs-1-vs-rest" approach, in such a way that each hyperplane is close to one class and far away from the other

class, while each training pattern is represented by ellipsoids instead of reduced convex hulls (the traditional SVM approach). The use of ellipsoids leads to SOCP formulations, conferring robustness to the solution.

The general description of the robust framework for maximum-margin classifiers based on second-order cones is provided in Section 3.1. The linear formulation of the proposed Twin-KSOCP method is presented in Section 3.2. The dual form of our proposal is provided in Section 3.3, where its geometrical properties are discussed. Finally, the kernel-based Twin-KSOCP method is described in Section 3.4.

3.1 Robust framework for maximum-margin classifiers

This section describes the robust framework based on conic constraints presented in [28]. Let $\mathbf{X}_k$ be a random vector that generates the samples of class k, with mean $\boldsymbol{\mu}_k \in \Re^n$, and covariance matrix Σ_k, for $k = 1, \ldots, K$, where $\Sigma_k \in \Re^{n\times n}$ are symmetric positive semi-definite matrices. Let us denote a family of distributions which have a common mean and covariance by $\mathbf{X} \sim (\boldsymbol{\mu}, \Sigma)$. In order to find a hyperplane that maximizes the margin of classification given by the moments of class conditional densities, [28] proposed the following probabilistic constraint:

$$\Pr\{\mathbf{w}^\top \mathbf{X}_k + b \geq 1\} \geq \eta_k, \tag{13}$$

with $\eta_k \in (0, 1)$ a predefined parameter that controls the misclassification rates for each class k. We want to classify each training pattern k correctly, up to the rate η_k, even for the worst data distribution. To accomplish this, the probabilistic constraint (13) can be replaced with its robust counterpart:

$$\inf_{\mathbf{X}_k \sim (\boldsymbol{\mu}_k, \Sigma_k)} \Pr\{\mathbf{w}^\top \mathbf{X}_k + b \geq 1\} \geq \eta_k. \tag{14}$$

Equation (14) can be cast into a second-order cone (SOC) constraint[1] by the application of the Chebyshev inequality [20, Lemma 1]. Equation (14) then becomes:

$$\mathbf{w}^\top \boldsymbol{\mu}_k + b \geq 1 + \kappa_k \|S_k^\top \mathbf{w}\|, \tag{15}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}$, $\Sigma_k = S_k S_k^\top$, for $k = 1, \ldots, K$. For example, the Cholesky factorization can be used to compute S_k from Σ_k.

Originally developed for binary classification [25, 28], this robust setting was extended to multiclass learning in [22, 23] for the One-versus-All and One-versus-One approaches, and two different "all-together" MC strategies. Notice that these methods are completely different compared with our twin formulation. Another difference is that our proposal is extended as a kernel method, in contrast with the models proposed in [22, 23].

[1] Recall that an SOC constraint on variable $\mathbf{x} \in \Re^n$ has the form $\|D\mathbf{x} + \mathbf{b}\| \leq \mathbf{c}^\top \mathbf{x} + d$, where $d \in \Re$, $\mathbf{c} \in \Re^n$, $\mathbf{b} \in \Re^m$, $D \in \Re^{m\times n}$ are given.

3.2 Linear Twin-KSOCP formulation

Let $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ be random vectors that generate the samples associated with matrices A, B, and C, respectively. Let us also denote by $\boldsymbol{\mu}_k \in \Re^n$ and $\Sigma_k \in \Re^{n\times n}$ the mean and the covariance matrix associated with random vector $\mathbf{X}_k$ for $k = 1, 2, 3$, where Σ_k are symmetric positive semi-definite matrices. In order to obtain a robust version of the Twin-KSVC method (Formulation (6)–(7)), we consider the following quadratic chance-constrained programming problems:

$$\begin{array}{ll} \min_{\mathbf{w}_1, b_1} & \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) \\ \text{s.t.} & \inf_{\mathbf{X}_2 \sim (\boldsymbol{\mu}_2, \Sigma_2)} \Pr\{\mathbf{w}_1{}^\top \mathbf{X}_2 + b_1 \leq -1\} \geq \eta_1, \\ & \inf_{\mathbf{X}_3 \sim (\boldsymbol{\mu}_3, \Sigma_3)} \Pr\{\mathbf{w}_1{}^\top \mathbf{X}_3 + b_1 \leq -(1-\epsilon)\} \geq \eta_2, \end{array}$$

and

$$\begin{array}{ll} \min_{\mathbf{w}_2, b_2} & \frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \frac{\theta_2}{2}(\|\mathbf{w}_2\|^2 + b_2^2) \\ \text{s.t.} & \inf_{\mathbf{X}_1 \sim (\boldsymbol{\mu}_1, \Sigma_1)} \Pr\{\mathbf{w}_2{}^\top \mathbf{X}_1 + b_2 \geq 1\} \geq \eta_3, \\ & \inf_{\mathbf{X}_3 \sim (\boldsymbol{\mu}_3, \Sigma_3)} \Pr\{\mathbf{w}_2{}^\top \mathbf{X}_3 + b_2 \geq (1-\epsilon)\} \geq \eta_4, \end{array}$$

where $\theta_1, \theta_2, \epsilon > 0$. The parameters $\eta_1, \ldots, \eta_4$ have an interpretation similar to the SOCP-SVM formulation, with values in $(0, 1)$. In particular, η_1 (η_3) aims at classifying the positive (negative) class correctly, while η_2 and η_4 aim at mapping the remaining observations into a zone between both hyperplanes.

Thanks to an appropriate application of the multivariate Chebyshev inequality, the above problems can be stated as the following quadratic SOCP problems (Twin-KSOCP formulation):

$$\begin{array}{ll} \min_{\mathbf{w}_1, b_1} & \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) \\ \text{s.t.} & -\mathbf{w}_1^\top \boldsymbol{\mu}_2 - b_1 \geq 1 + \kappa_1 \|S_2^\top \mathbf{w}_1\|, \\ & -\mathbf{w}_1^\top \boldsymbol{\mu}_3 - b_1 \geq 1 - \epsilon + \kappa_2 \|S_3^\top \mathbf{w}_1\|, \end{array} \tag{16}$$

and

$$\begin{array}{ll} \min_{\mathbf{w}_2, b_2} & \frac{1}{2}\|B\mathbf{w}_2 + \mathbf{e}_2 b_2\|^2 + \frac{\theta_2}{2}(\|\mathbf{w}_2\|^2 + b_2^2) \\ \text{s.t.} & \mathbf{w}_2^\top \boldsymbol{\mu}_1 + b_2 \geq 1 + \kappa_3 \|S_1^\top \mathbf{w}_2\|, \\ & \mathbf{w}_2^\top \boldsymbol{\mu}_3 + b_2 \geq 1 - \epsilon + \kappa_4 \|S_3^\top \mathbf{w}_2\|, \end{array} \tag{17}$$

where $\Sigma_i = S_i S_i^\top$ for $i = 1, 2, 3$, and $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$ for $i = 1, 2, 3, 4$.

Remark 1 Note that the objective functions of problems (16)–(17) can be written compactly as

$$\frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) = \frac{1}{2}\mathbf{v}_1^\top (H^\top H + \theta_1 I)\mathbf{v}_1, \tag{18}$$

and

$$\frac{1}{2}\|B\mathbf{w}_2+\mathbf{e}_2 b_2\|^2+\frac{\theta_2}{2}(\|\mathbf{w}_2\|^2+b_2^2)=\frac{1}{2}\mathbf{v}_2^\top(G^\top G+\theta_2 I)\mathbf{v}_2, \tag{19}$$

respectively, where

$$\mathbf{v}_k=[\mathbf{w}_k;b_k]\in\Re^{n+1},\ H=[A\ \mathbf{e}_1]\in\Re^{m_1\times(n+1)},\quad G=[B\ \mathbf{e}_2]\in\Re^{m_2\times(n+1)}. \tag{20}$$

Then, by introducing the new variables t_1, t_2 and the constraints

$$\|(H^\top H+\theta_1 I)^{1/2}\mathbf{v}_1\|\le t_1,\quad \|(G^\top G+\theta_2 I)^{1/2}\mathbf{v}_2\|\le t_2,$$

the problems (16) and (17) can be cast into linear SOCP problems with three SOC constraints each.

The decision function is similar to the one used for the Twin-KSVC method (cf. (8)).

3.3 A. Dual formulation of Twin-KSOCP and geometric interpretation

In this section, we present the dual formulation of Twin-KSOCP (Formulations (16) and (17)), and provide geometrical insights for the method.

The following Proposition gives the dual formulation of problems (16)–(17):

Proposition 1 *Let us denote by* $\hat{H}=H^\top H+\theta_1 I$, $\hat{G}=G^\top G+\theta_2 I$, $\hat{\mathbf{z}}_i=[\mathbf{z}_i;1]\in\Re^{n+1}$, $\hat{\mathbf{p}}_i=[\mathbf{p}_i;1]\in\Re^{n+1}$, *for* $i=1,2$. *Then, the duals of the problems* (16)–(17) *are given by*

$$\begin{array}{ll}\max_{\mathbf{z}_i,\mathbf{u}_i} & \frac{1}{2}\frac{h_1(\hat{\mathbf{z}}_1,\hat{\mathbf{z}}_2,\hat{H})}{h_2(\hat{\mathbf{z}}_1,\hat{\mathbf{z}}_2,\hat{H})}\\ s.t. & \mathbf{z}_1\in\mathbf{B}(\boldsymbol{\mu}_2,S_2,\kappa_1),\quad \mathbf{z}_2\in\mathbf{B}(\boldsymbol{\mu}_3,S_3,\kappa_2),\end{array} \tag{21}$$

and

$$\begin{array}{ll}\max_{\mathbf{p}_i,\mathbf{u}_i} & \frac{1}{2}\frac{h_1(\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\hat{G})}{h_2(\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\hat{G})}\\ s.t. & \mathbf{p}_1\in\mathbf{B}(\boldsymbol{\mu}_1,S_1,\kappa_3),\quad \mathbf{p}_2\in\mathbf{B}(\boldsymbol{\mu}_3,S_3,\kappa_4),\end{array} \tag{22}$$

where

$$h_1(\mathbf{z}_1,\mathbf{z}_2,H)=\|H^{-1/2}(\mathbf{z}_2-(1-\epsilon)\mathbf{z}_1)\|^2,$$

$$h_2(\mathbf{z}_1,\mathbf{z}_2,H)=(\|H^{-1/2}\mathbf{z}_1\|\|H^{-1/2}\mathbf{z}_2\|)^2-(\mathbf{z}_1^{\top H^{-1}}\mathbf{z}_2)^2,$$

and

$$\mathbf{B}(\boldsymbol{\mu},S,\kappa)=\{\mathbf{z}:\mathbf{z}=\boldsymbol{\mu}+\kappa S\mathbf{u},\|\mathbf{u}\|\le 1\}.$$

The set $\mathbf{B}(\boldsymbol{\mu},S,\kappa)$ *denotes an ellipsoid centered at* $\boldsymbol{\mu}$ *whose shape is determined by* S, *and size by* κ.

The proof of Proposition 1 is presented in Appendix. The optimization problems that result from our proposal are fractional programming problems, and they can be solved, for instance, by Dinkelbach-type algorithms [11]. Additionally, from this result we obtain that the dual problems (21) and (22) can be seen as the maximization of the ratio between two functions over two ellipsoids. These ellipsoids define the two twin hyperplanes, and, subsequently, the classification rule for the proposed Twin-KSOCP method.

The following remark associates the primal and dual variables of the Twin-KSOCP formulation, which is relevant since we can solve the dual formulations and then obtain the two hyperplanes.

Remark 2 Once the Problem (21) is resolved, we can derive the optimal solution $\mathbf{v}_1^*=[\mathbf{w}_1^*;b_1^*]$ of Problem (16) as follows (cf. (A.34), (A.36)):

$$\mathbf{v}_1^*=-\hat{H}^{-1}(\lambda_1\hat{\mathbf{z}}_1+\lambda_2\hat{\mathbf{z}}_2), \tag{23}$$

where

$$\lambda_1=\frac{\hat{\mathbf{z}}_2^\top\hat{H}^{-1}\hat{\mathbf{z}}_2-(1-\epsilon)\hat{\mathbf{z}}_1^\top\hat{H}^{-1}\hat{\mathbf{z}}_2}{h_2(\hat{\mathbf{z}}_1,\hat{\mathbf{z}}_2,\hat{H})},$$
$$\lambda_2=\frac{(1-\epsilon)\hat{\mathbf{z}}_1^\top\hat{H}^{-1}\hat{\mathbf{z}}_1-\hat{\mathbf{z}}_1^\top\hat{H}^{-1}\hat{\mathbf{z}}_2}{h_2(\hat{\mathbf{z}}_1,\hat{\mathbf{z}}_2,\hat{H})}. \tag{24}$$

Similarly, we can derive the optimal solution $\mathbf{v}_2^*=[\mathbf{w}_2^*;b_2^*]$ of Problem (17) as follows:

$$\mathbf{v}_2^*=\hat{G}^{-1}(\alpha_1\hat{\mathbf{p}}_1+\alpha_2\hat{\mathbf{p}}_2), \tag{25}$$

where

$$\alpha_1=\frac{\hat{\mathbf{p}}_2^\top\hat{G}^{-1}\hat{\mathbf{p}}_2-(1-\epsilon)\hat{\mathbf{p}}_1^\top\hat{G}^{-1}\hat{\mathbf{p}}_2}{h_2(\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\hat{G})},$$
$$\alpha_2=\frac{(1-\epsilon)\hat{\mathbf{p}}_1^\top\hat{G}^{-1}\hat{\mathbf{p}}_1-\hat{\mathbf{p}}_1^\top\hat{G}^{-1}\hat{\mathbf{p}}_2}{h_2(\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\hat{G})}. \tag{26}$$

3.4 Kernel-based Twin SOCP-KSVC formulation

In this section, we extend the proposed Twin-KSOCP to kernel functions in order to obtain non-linear classifiers. For this, we first notice that the weight vectors for each one of the twin hyperplanes can be written as $\mathbf{w}_k=\mathbb{X}\,\mathbf{s}_k+M\mathbf{r}_k$, where M is a matrix whose columns (as vectors) are orthogonal to the training data points; $\mathbb{X}=[A^\top\ B^\top\ C^\top]$ is the data matrix defined in Section 2.5; and $\mathbf{s}_k$, $\mathbf{r}_k$ are vectors of combining coefficients with the appropriate dimensions.

On the other hand, the empirical estimates of the mean $\boldsymbol{\mu}_k$ and covariance Σ_k are given by

$$\hat{\boldsymbol{\mu}}_1=\frac{1}{m_1}A^\top\mathbf{e}_1,\quad \hat{\boldsymbol{\mu}}_2=\frac{1}{m_2}B^\top\mathbf{e}_2,\quad \hat{\boldsymbol{\mu}}_3=\frac{1}{m_3}C^\top\mathbf{e}_3,$$
$$\hat{\Sigma}_j=S_jS_j^\top,\ j=1,2,3,$$

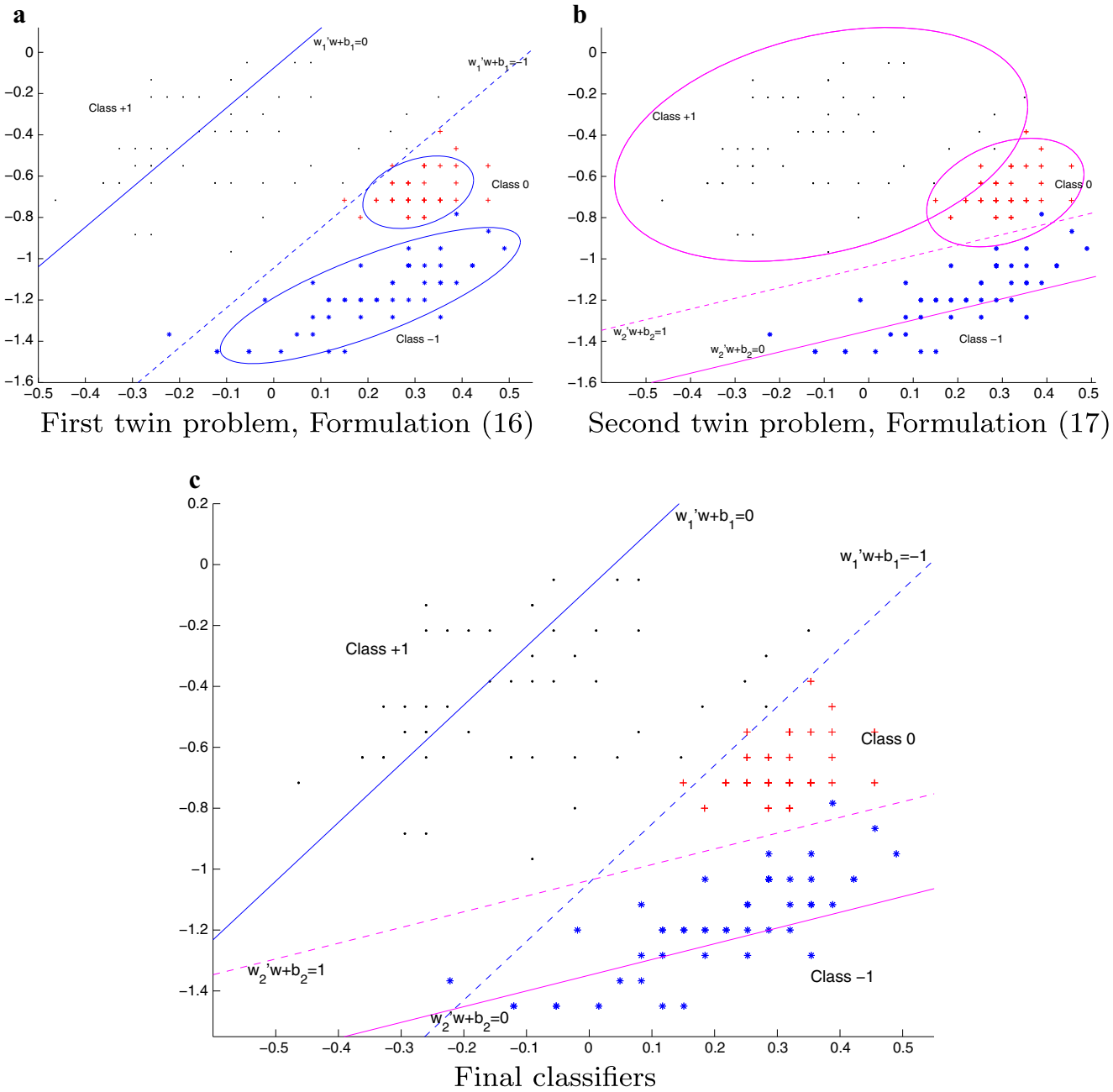

Fig. 1 Geometric interpretation for Twin-KSOCP

where

$$S_1 = \frac{1}{\sqrt{m_1}}(A^\top - \hat{\boldsymbol{\mu}}_1 \mathbf{e}_1^\top), \quad S_2 = \frac{1}{\sqrt{m_2}}(B^\top - \hat{\boldsymbol{\mu}}_2 \mathbf{e}_2^\top),$$
$$S_3 = \frac{1}{\sqrt{m_3}}(C^\top - \hat{\boldsymbol{\mu}}_3 \mathbf{e}_3^\top).$$

Then,

$$\mathbf{w}_k^\top \boldsymbol{\mu}_j = \mathbf{s}_k^\top \mathbf{g}_j, \ \mathbf{w}_k^\top \Sigma_j \mathbf{w}_k = \mathbf{s}_k^\top \Xi_j \mathbf{s}_k, \ k = 1, 2, \quad j = 1, 2, 3, \ j \neq k,$$

where

$$\mathbf{g}_j = \frac{1}{m_j} \begin{bmatrix} \mathbf{K}_{1j}\mathbf{e}_j \\ \mathbf{K}_{2j}\mathbf{e}_j \\ \mathbf{K}_{3j}\mathbf{e}_j \end{bmatrix},$$

$$\Xi_j = \frac{1}{m_j} \begin{bmatrix} \mathbf{K}_{1j} \\ \mathbf{K}_{2j} \\ \mathbf{K}_{3j} \end{bmatrix} \left(I_{m_j} - \frac{1}{m_j} \mathbf{e}_j \mathbf{e}_j^\top \right) \left[\mathbf{K}_{1j}^\top \ \mathbf{K}_{2j}^\top \ \mathbf{K}_{3j}^\top \right],$$

with $\mathbf{K}_{11} = AA^\top$, $\mathbf{K}_{12} = \mathbf{K}_{21}^\top = AB^\top$, $\mathbf{K}_{13} = \mathbf{K}_{31}^\top = AC^\top$, $\mathbf{K}_{22} = BB^\top$, $\mathbf{K}_{23} = \mathbf{K}_{32}^\top = BC^\top$, $\mathbf{K}_{33} = CC^\top$ matrices whose elements are inner products of data points.

Table 1 Number of examples, number of variables and number of classes for all data sets

Dataset	#examples	#variables	#classes
Iris	150	4	3
Hayes-Roth	160	4	3
Wine	178	13	3
Glass	214	13	6
Led7digit	500	7	10
Vowel	528	12	11
Fish	762	12	3
Segment	2310	19	7
Waveform	5000	21	3

For instance, the entry (l, s) for the matrix $\mathbf{K}_{kk'}$ is the following $(\mathbf{K}_{kk'})_{ls} = (\mathbf{x}_l^k)^\top \mathbf{x}_s^{k'}$. Hence, in order to obtain a kernel formulation for the problems (16) and (17), we replace the inner product above by any function $\mathcal{K} : \Re^n \times \Re^n \to \Re$, satisfying Mercer's condition [27]. That is, the products $(\mathbf{x}_l^k)^\top \mathbf{x}_s^{k'}$ are replaced by $(\mathbf{K}_{kk'})_{ls} = \mathcal{K}(\mathbf{x}_l^k, \mathbf{x}_s^{k'})$. Thus, we obtain the following optimization problems (kernel-based Twin-KSOCP):

$$\begin{array}{ll} \min_{\mathbf{s}_1, b_1} & \frac{1}{2}\|\mathbf{K}_{1\bullet}\mathbf{s}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{s}_1\|^2 + b_1^2) \\ \text{s.t.} & -\mathbf{s}_1^\top \mathbf{g}_2 - b_1 \geq 1 + \kappa_1 \|\Lambda_2^\top \mathbf{s}_1\|, \\ & -\mathbf{s}_1^\top \mathbf{g}_3 - b_1 \geq 1 - \epsilon + \kappa_2 \|\Lambda_3^\top \mathbf{s}_1\|, \end{array} \tag{27}$$

and

$$\begin{array}{ll} \min_{\mathbf{s}_2, b_2} & \frac{1}{2}\|\mathbf{K}_{2\bullet}\mathbf{s}_2 + \mathbf{e}_2 b_2\|^2 + \frac{\theta_2}{2}(\|\mathbf{s}_2\|^2 + b_2^2) \\ \text{s.t.} & \mathbf{s}_2^\top \mathbf{g}_1 + b_2 \geq 1 + \kappa_3 \|\Lambda_1^\top \mathbf{s}_2\|, \\ & \mathbf{s}_2^\top \mathbf{g}_3 + b_2 \geq 1 - \epsilon + \kappa_4 \|\Lambda_3^\top \mathbf{s}_2\|, \end{array} \tag{28}$$

where $\Xi_j = \Lambda_j \Lambda_j^\top$, for $j = 1, 2, 3$. Then, the solutions for Problems (27) and (28) lead to the following kernel-based surfaces:

$$\mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_1 + b_1 = 0, \quad \mathcal{K}(\mathbf{x}, \mathbb{X})\mathbf{s}_2 + b_2 = 0, \tag{29}$$

where, for a given $\mathbf{x} \in \Re^n$, the row vector $\mathcal{K}(\mathbf{x}, \mathbb{X})$ is defined by

$$\mathcal{K}(\mathbf{x}, \mathbb{X}) = [\mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet 1}), \mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet 2}), \ldots, \mathcal{K}(\mathbf{x}, \mathbb{X}_{\bullet m})],$$

with $\mathbb{X}_{\bullet j} \in \Re^n$ denoting the j-th column of the matrix $\mathbb{X}$. The decision function is similar to (12).

4 Experimental results

We applied the proposed Twin-KSOCP approach in its linear and kernel-based version to nine well-known benchmark datasets for multi-class classification. We studied other alternative multi-class SVM formulations described in Section 2 (MC-SVM, OVO-SVM, OVA-SVM, OVA-TWSVM, and Twin KSVC) for comparison purposes.

In addition, we studied three highly-optimized SVM approximations for large-scale multiclass classification: Pegasos, Adaptive Multi-Hyperplane Machine (AMM), and Budgeted Stochastic Gradient Descent (BSGD) [12]. The first approach alternates stochastic sub-gradient descent steps and projection steps iteratively for efficient SVM training [36]. The AMM approach construct non-linear classifiers by using multiple linear hyperplanes, as presented in [37]. The third method, BSGD, uses stochastic gradient descent to update the number of support vectors [38]. The first method is only available as a linear method, while the remaining two are used as kernel-based approaches [12].

This section is organized as follows: First, an illustrative example is provided in Section 4.1 in order to illustrate the functioning of our proposal with a two-dimensional toy dataset. Next, a description of the benchmark datasets used in this work is provided in Section 4.2. Section 4.3 presents a summary of the performance obtained for the proposed and alternative approaches. Finally, the running times for each method are reported in Section 4.4.

Table 2 Performance summary for different classification approaches

	Iris	Hayes-Roth	Wine	Glass	Led7digit	Vowel	Fish	Segment	Waveform
MC-SVM	96.0	57.9	99.0	57.3	**75.2**	72.1	69.7	88.2	**87.2**
OVA-SVM	94.7	61.5	98.6	60.7	74.1	56.4	74.4	92.7	87.0
OVO-SVM	**98.0**	64.9	98.6	66.1	74.3	**90.0**	**80.0**	**95.5**	87.0
OVA-TWSVM	93.3	65.4	99.0	58.7	74.0	58.3	75.1	93.2	87.0
Twin-KSVC	**98.0**	60.1	98.9	47.2	73.5	81.9	76.6	94.7	86.0
Twin-KSOCP	96.7	**69.5**	**99.5**	**70.6**	73.8	69.0	76.6	93.3	85.7
Pegasos	97.3	56.8	98.3	51.4	66.8	51.7	67.3	81.5	84.4

Linear classifiers

Table 3 Performance summary for different classification approaches

	Iris	Hayes-Roth	Wine	Glass	Led7digit	Vowel	Fish	Segment	Waveform
MC-SVM	97.3	87.8	99.0	71.4	**75.9**	99.0	83.2	98.3	87.0
OVA-SVM	97.3	87.2	99.5	71.8	74.2	**99.6**	81.6	**97.5**	**87.2**
OVO-SVM	**98.0**	87.7	99.0	72.2	74.7	**99.6**	82.6	97.4	87.0
OVA-TWSVM	**98.0**	87.1	98.4	71.7	71.4	98.5	**87.7**	96.7	86.5
Twin-KSVC	97.3	88.5	99.3	62.7	71.8	87.1	81.5	95.2	84.5
Twin-KSOCP	**98.0**	**89.1**	**100.0**	**74.0**	74.5	99.5	82.5	95.7	86.2
AMM	96.7	49.7	98.3	57.0	66.4	61.7	73.2	83.9	86.2
BSGD	96.0	53.8	96.7	73.3	60.0	98.3	62.9	95.9	85.5

Kernel-based classifiers

4.1 Illustrative example

Figure 1 illustrates the geometrical interpretation of the proposed Twin-KSOCP method in its linear version, using a two-dimensional toy data set with three classes.

Figure 1a shows the solution of the first twin problem, Formulation (16). This SOCP problem aims at correctly representing Class +1, and therefore the result of this model is a hyperplane over this class (the solid line), and another one representing the remaining classes, Class 0 and Class -1 (the dashed line), represented by the two ellipsoids.

Figure 1b shows the solution of the second twin problem, Formulation (17). This model studies the relationship between Class -1 and the remaining observations. The resulting output is a first hyperplane over this class (the solid line), and a second one representing classes 0 and +1 (the dashed line).

Finally, Fig. 1c illustrates the four hyperplanes obtained from both twin problems, allowing the illustration of the three-way decision rule: a new sample is classified as Class +1 if its evaluation using the first hyperplane is positive enough ($\mathbf{w}_1^\top \mathbf{x} + b_1 > -1 + \epsilon$, see (8)), as Class -1 if its evaluation using the second hyperplane is negative enough ($\mathbf{w}_2^\top \mathbf{x} + b_2 < 1 - \epsilon$), or as Class 0 otherwise.

4.2 Datasets and experimental settings

We studied eight datasets from the UCI Machine Learning Repository [3]: Iris, Wine, Glass, Vowel, Hayes-Roth, Led7Digit, Waveform, and Segment; and one dataset from a previous project in the classification of fish schools (Fish, see [4] for more details). Table 1 summarizes the relevant information for each data set:

We used 10-fold cross-validation for model selection purposes, and balanced accuracy was used as the main performance metric. This measure is computed as the average class accuracy (also known as *recall*) among all classes. We explored the following values for the hyperparameters C (traditional SVM approaches), c_i, $i = \{1, 2, 3, 4\}$ (twin SVM classifiers), θ_1 and θ_2 (Twin-KSOCP), and σ (kernel-based methods, where Gaussian kernel was used): $\{2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7\}$. We imposed $c_1 = c_3$ and $c_2 = c_4$ for twin SVM classifiers, and $\theta_1 = \theta_2$ for kernel-based Twin-KSOCP. For parameter ϵ we explore the following values: $\{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$.

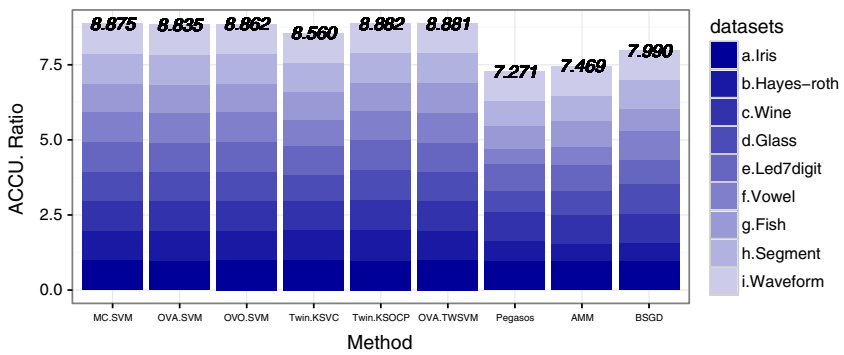

Fig. 2 Sum of accuracy ratios for all methods

Table 4 Holm's post-hoc test for pairwise comparisons

Method	Mean Rank	Mean AUC	p value	$\alpha/(k-i)$	Action
Twin-KSOCP	3.00	88.83	-	-	not reject
OVO-SVM	3.05	88.69	0.9657	0.0500	not reject
MC-SVM	3.27	88.79	0.8296	0.0250	not reject
OVA-SVM	3.44	88.43	0.7306	0.0167	not reject
OVA- TWSVM	4.11	88.86	0.3894	0.0125	not reject
Twin-KSVC	5.27	85.76	0.0777	0.0100	not reject
BSGD	7.22	80.27	0.0011	0.0083	reject
AMM	7.67	74.79	0.0003	0.0071	reject
Pegasos	7.94	72.83	0.0001	0.0062	reject

4.3 Summary of results

Next, a summary of the results is presented for all nine data sets. Tables 2 and 3 show the best performance for each method in terms of balanced accuracy for both linear and kernel-based classifiers, respectively. The best strategy for each data set is highlighted in bold type.

In Table 2, we observe that OVO-SVM and the proposed method (Twin-KSOCP, linear version) are the best approaches, achieving the best performance in four and three of the nine data sets, respectively, while the remaining methods perform best only once, or not at all. For the kernel-based methods (Table 3), the proposed Twin-KSOCP has the best overall performance, achieving the highest accuracy in four out of the nine data sets. It is important to notice that the proposed Twin-KSOCP always achieves a better performance compared to Twin-KSVC, demonstrating the effectiveness of the robust framework, especially for the kernel-based formulation.

A comparison between linear and kernel-based classifiers (Tables 2 and 3, respectively) leads to a noticeable conclusion: the kernel-based versions of each classifier lead to important gains in terms of performance. In particular, the use of the Gaussian kernel outperforms the linear formulation for datasets Hayes-Roth, Glass, Vowel, and Fish. For the other data sets the results are either close to 100%, and therefore there is little room for improvement (Iris and Wine), or the gain is only marginal (Led7digit, Segment, and Waveform).

In order to assess the overall performance among methods, we follow the strategy proposed in [14]: first, the highest accuracy between linear and kernel-based classifiers is computed for each method. Then, a relative performance, called *accuracy ratio*, is obtained by dividing the balanced accuracy of each technique by the highest one among all the strategies compared. The best method in a given data set will have an accuracy ratio of 1. Finally, the sum of all accuracy ratios for all data sets provides a good indicator of the best overall performance and robustness for each method. Figure 2 presents this measure for all the techniques used in this work.

In Fig. 2, we observe that the proposed approach has the best overall performance, followed closely by OVA-TWSVM, demonstrating the virtues of our approach and twin SVM classification for multiclass applications.

Next, we use the Holm's test to assess statistical significance. This test was proposed by [16], and recommended in [10] for comparing among various machine learning methods. The idea is to compute the average rank for each technique, and perform pairwise comparisons between each

Table 5 Average running times, in seconds, for all datasets

	Iris	Hayes-Roth	Wine	Glass	Led7digit	Vowel	Fish	Segment	Waveform
MC-SVM	0".48	0".48	0".56	6".33	441".73	390".42	23".85	14627".13	6323".20
OVA-SVM	0".37	0".38	0".43	1".16	0".82	29".53	0".58	59".77	14".78
OVO-SVM	0".20	0".25	0".25	0".90	4".66	3".97	0".37	9".15	5".22
OVA-TWSVM	0".17	0".05	0".08	0".35	1".54	6".59	2".68	48".81	24".44
Twin-KSVC	2".10	0".72	0".41	18".06	73".84	255".52	15".53	2357".98	1623".68
Twin-KSOCP$_l$	0".60	0".50	0".51	2".61	27".62	42".06	2".29	17".77	3".58
Twin-KSOCP$_k$	2".51	2".73	2".89	21".55	235".63	310".36	33".16	1918".88	1889".75
Pegasos	0".05	0".03	0".04	0".04	0".06	0".05	0".07	0".05	0".06
AMM	0".05	0".05	0".07	0".08	0".08	0".11	0".05	0".09	0".08
BSGD	0".02	0".02	0".02	0".08	0".16	0".73	0".93	2".68	6".16

approach and the one with the lowest rank. The results for this analysis are presented in Table 4.

In Table 4, we can see that the standard and twin SVM methods outperform the highly optimized approaches using a significance level $\alpha = 0.05$, where $i = 1$ (best approach, Twin-KSOCP) and $k = 2, 3, ..., 8$ is the ranking of the remaining methods. Although our proposal has the best overall performance, we conclude that no method outperforms the others statistically in terms of maximum AUC among all the subsets of variables for the six most relevant comparisons.

4.4 Running times

In Table 5, a comparison in terms of the average running times is reported. The experiments were performed on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, a i7-2620M processor with 2.70 GHz, and using Microsoft Windows 8.1 Operating System (64-bits). We used the SeDuMi Matlab Toolbox [33] for the proposed method; Budgeted SVM toolbox [12] for Pegasos, AMM, and BSGD; the codes provided by Yuan-Hai Shao, author of Twin-Bounded SVM [32], which are publicly available in http://www.optimal-group.org/, for the twin SVM methods; and the spider toolbox [40] and LIBSVM [8] were used for the standard multiclass SVM approaches. The Twin-KSVC method was implemented by using the successive overrelaxation (SOR) technique, as suggested in [32] for binary classification.

On Table 5 we observe that our approach in its linear form (Twin-KSOCP$_l$) is consistently faster than Twin-KSVC, achieving similar running times compared to standard SVM methods. For the kernel version (Twin-KSOCP$_k$), however, higher training times are obtained, which are relatively similar compared to Twin-KSVC.

5 Conclusions

In this work, a novel second-order cone programming formulation for multiclass classification is proposed. This method extends the ideas of twin SVM for binary labels [17] and Twin-KSVC [42] for multiclass labels to second-order cones. The proposed method follows a one-vs.-one-vs.-rest competition scheme [2], where data points are classified either as class "+1", class "-1", or neither of them (class "0"). Each classification problem constructs two nonparallel hyperplanes in such a way that each one is close to one of the two studied classes, and as far as possible from the other. Instead of using the reduced convex hulls to maximize the separation margin (traditional SVM approach), the proposed SOCP formulation constructs ellipsoids based on the mean and covariance matrix of each training pattern [22, 28]. Proposed as a linear classification method, Twin-KSOCP is further extended to kernel-based formulation, while the dual is computed in order to study its geometrical interpretation.

The main contribution of our proposal is a robust framework for multiclass classification. The probabilistic framework is designed to classify all classes correctly, at least to a predefined class recall η, even for the worst data distribution. Empirically, our proposed method achieved best overall classification performance within tractable training times. The proposal can also be used as a kernel method, conferring flexibility to the approach. A comparison between various standard SVM and twin SVM methods for multiclass learning using nine benchmark data sets demonstrates the virtues of our strategy, in particular when using Gaussian kernels. Finally, duality theory provides interesting geometrical properties.

There are interesting future developments that can be derived from this work. First, there are various strategies for modelling the margin maximization in SVM that can be used for twin multiclass classification. Two examples are the use of flexible convex hulls, and affine convex hulls [45]. Secondly, this proposal can be further extended to high-dimensional data sets, where feature selection is a necessary step for adequate prediction [21]. Finally, another issue that often arises with high-dimensional tasks is the "class-imbalance problem" [7], in which one or more categories are under-represented in the data set. In this context, the proposed framework specifies different error rates for each training pattern, providing an interesting approach for class-imbalance classification. Although this idea has already been studied in binary classification [24], the proposed method can be suitable for a multiclass task with skewed label distribution.

Acknowledgments The first author was supported by FONDECYT project 1160894, the second was supported by FONDECYT projects 1140831 and 1160738, and the third author was supported by FONDECYT project 1130905. This research was partially funded by the Complex Engineering Systems Institute, ISCI (ICM-FIC: P05-004-F, CONICYT: FB0816).

Appendix: Dual formulation of Twin-KSOCP and geometric interpretation

Proof of Proposition 1

The Lagrangian function associated with Problem (16) is given by

$$\begin{aligned} L(\mathbf{w}_1, b_1, \lambda_1, \lambda_2) = & \frac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \frac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) \\ & + \lambda_1(\mathbf{w}_1^\top \boldsymbol{\mu}_2 + b_1 + 1 + \kappa_1 \|S_2^\top \mathbf{w}_1\|) \\ & + \lambda_2(\mathbf{w}_1^\top \boldsymbol{\mu}_3 + b_1 + 1 - \epsilon + \kappa_2 \|S_3^\top \mathbf{w}_1\|), \end{aligned}$$

where $\lambda_1, \lambda_2 \geq 0$. Since $\|\mathbf{v}\| = \max_{\|\mathbf{u}\|\leq 1} \mathbf{u}^\top \mathbf{v}$ holds for any $\mathbf{v} \in \Re^n$, we can rewrite the Lagrangian as follows:

$$L(\mathbf{w}_1, b_1, \lambda_1, \lambda_2) = \max_{\mathbf{u}} \{L_1(\mathbf{w}_1, b_1, \lambda_1, \lambda_2, \mathbf{u}_1, \mathbf{u}_2) : \|\mathbf{u}_i\| \leq 1,\ i = 1, 2\},$$

with L_1 given by

$$\begin{aligned} L_1(\mathbf{w}_1, b_1, \lambda_1, \lambda_2, \mathbf{u}_1, \mathbf{u}_2) &= \tfrac{1}{2}\|A\mathbf{w}_1 + \mathbf{e}_1 b_1\|^2 + \tfrac{\theta_1}{2}(\|\mathbf{w}_1\|^2 + b_1^2) \\ &+\lambda_1(\mathbf{w}_1^\top \boldsymbol{\mu}_2 + b_1 + 1 + \kappa_1 \mathbf{w}_1^\top S_2 \mathbf{u}_1) \\ &+\lambda_2(\mathbf{w}_1^\top \boldsymbol{\mu}_3 + b_1 + 1 - \epsilon + \kappa_2 \mathbf{w}_1^\top S_3 \mathbf{u}_2). \end{aligned} \tag{A.30}$$

Thus, Problem (16) can be equivalently written as

$$\min_{\mathbf{w}_1, b_1} \max_{\mathbf{u}_1, \mathbf{u}_2, \lambda_1, \lambda_2} \{L_1(\mathbf{w}_1, b_1, \lambda_1, \lambda_2, \mathbf{u}_1, \mathbf{u}_2) : \|\mathbf{u}_i\| \leq 1, \lambda_i \geq 0,\ i = 1, 2\}.$$

Hence, the dual problem of (16) is given by

$$\max_{\mathbf{u}_1, \mathbf{u}_2, \lambda_1, \lambda_2} \min_{\mathbf{w}_1, b_1} \{L_1(\mathbf{w}_1, b_1, \lambda_1, \lambda_2, \mathbf{u}_1, \mathbf{u}_2) : \|\mathbf{u}_i\| \leq 1, \lambda_i \geq 0,\ i = 1, 2\}. \tag{A.31}$$

The above expression allows the construction of the dual formulation. A detailed description of this procedure can be found in [26]. The computation of the first order condition for the inner optimization task (the minimization problem) yields to

$$\begin{aligned} \nabla_{\mathbf{w}_1} L_1 &= A^\top(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + \theta_1 \mathbf{w}_1 + \lambda_1(\boldsymbol{\mu}_2 + \kappa_1 S_2 \mathbf{u}_1) \\ &\quad +\lambda_2(\boldsymbol{\mu}_3 + \kappa_2 S_3 \mathbf{u}_2) = 0, \end{aligned} \tag{A.32}$$

$$\nabla_{b_1} L_1 = \mathbf{e}_1^\top(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + \theta_1 b_1 + \lambda_1 + \lambda_2 = 0. \tag{A.33}$$

Let us denote by $\hat{\mathbf{z}}_1 = [\mathbf{z}_1; 1]$, $\hat{\mathbf{z}}_2 = [\mathbf{z}_2; 1] \in \Re^{n+1}$, with $\mathbf{z}_1 = \boldsymbol{\mu}_2 + \kappa_1 S_2 \mathbf{u}_1 \in \Re^n$, and $\mathbf{z}_2 = \boldsymbol{\mu}_3 + \kappa_2 S_3 \mathbf{u}_2 \in \Re^n$. Then the relations (A.33)–(A.33) can be written compactly as

$$(H^\top H + \theta_1 I)\mathbf{v}_1 + \lambda_1 \hat{\mathbf{z}}_1 + \lambda_2 \hat{\mathbf{z}}_2 = 0,$$

where $\mathbf{v}_1 = [\mathbf{w}_1; b_1]$ and $H = [A\ \mathbf{e}_1]$. Since the symmetric matrix $\hat{H} = H^\top H + \theta_1 I \in \Re^{n+1 \times n+1}$ is positive definite, for any $\theta_1 > 0$, the following relation can be obtained:

$$\mathbf{v}_1 = -\hat{H}^{-1}(\lambda_1 \hat{\mathbf{z}}_1 + \lambda_2 \hat{\mathbf{z}}_2). \tag{A.34}$$

Then, by replacing (A.33)–(A.33) in (A.30), and using the relations (18) and (A.34), the dual problem can be stated as follows:

$$\begin{aligned} \max_{\mathbf{z}_i, \mathbf{u}_i, \lambda_i} \quad & \lambda_1 + \lambda_2(1 - \epsilon) - \tfrac{1}{2}(\lambda_1 \hat{\mathbf{z}}_1 + \lambda_2 \hat{\mathbf{z}}_2)^\top \hat{H}^{-1}(\lambda_1 \hat{\mathbf{z}}_1 + \lambda_2 \hat{\mathbf{z}}_2) \\ \text{s.t.} \quad & \mathbf{z}_1 = \boldsymbol{\mu}_2 + \kappa_1 S_2 \mathbf{u}_1,\ \|\mathbf{u}_1\| \leq 1, \\ & \mathbf{z}_2 = \boldsymbol{\mu}_3 + \kappa_2 S_3 \mathbf{u}_2,\ \|\mathbf{u}_2\| \leq 1, \\ & \lambda_1, \lambda_2 \geq 0. \end{aligned} \tag{A.35}$$

Notice that the Hessian of the objective function of the above problem with respect to $\boldsymbol{\lambda} = [\lambda_1; \lambda_2] \in \Re^2$ is given by

$$H_z = \begin{pmatrix} \hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_1 & \hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 \\ \hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 & \hat{\mathbf{z}}_2^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 \end{pmatrix}.$$

Clearly, this matrix is symmetric positive definite. Then, the objective function of the dual problem (A.35) is strictly concave with respect to $\boldsymbol{\lambda}$, and it attains its maximum value at the solution of the following linear system:

$$H_z \begin{pmatrix} \lambda_1^* \\ \lambda_2^* \end{pmatrix} = \begin{pmatrix} 1 \\ 1 - \epsilon \end{pmatrix}.$$

This linear system has the following solution:

$$\begin{aligned} \lambda_1^* &= \frac{\hat{\mathbf{z}}_2^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 - (1 - \epsilon)\hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2}{\det(H_z)}, \\ \lambda_2^* &= \frac{(1 - \epsilon)\hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_1 - \hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2}{\det(H_z)}. \end{aligned} \tag{A.36}$$

Thus, the optimal value of Problem (A.35) (with respect to $\boldsymbol{\lambda}$) is given by

$$\frac{1}{2}(1 \quad 1 - \epsilon)(H_z)^{-1} \begin{pmatrix} 1 \\ 1 - \epsilon \end{pmatrix}, \tag{A.37}$$

where

$$(H_z)^{-1} = \frac{1}{\det(H_z)} \begin{pmatrix} \hat{\mathbf{z}}_2^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 & -\hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 \\ -\hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2 & \hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_1 \end{pmatrix}.$$

Then, the dual problem of (16) can be stated as follows:

$$\begin{aligned} \max_{\mathbf{z}_i, \mathbf{u}_i} \quad & \frac{1}{2} \frac{\|\hat{H}^{-1/2}(\hat{\mathbf{z}}_2 - (1-\epsilon)\hat{\mathbf{z}}_1)\|^2}{(\|\hat{H}^{-1/2}\hat{\mathbf{z}}_1\| \|\hat{H}^{-1/2}\hat{\mathbf{z}}_2\|)^2 - (\hat{\mathbf{z}}_1^\top \hat{H}^{-1} \hat{\mathbf{z}}_2)^2} \\ \text{s.t.} \quad & \mathbf{z}_1 \in \mathbf{B}(\boldsymbol{\mu}_2, S_2, \kappa_1), \quad \mathbf{z}_2 \in \mathbf{B}(\boldsymbol{\mu}_3, S_3, \kappa_2), \end{aligned} \tag{A.38}$$

where

$$\mathbf{B}(\boldsymbol{\mu}, S, \kappa) = \{\mathbf{z} : \mathbf{z} = \boldsymbol{\mu} + \kappa S \mathbf{u}, \|\mathbf{u}\| \leq 1\}. \tag{A.39}$$

Similarly, since the symmetric matrix $\hat{G} = G^\top G + \theta_2 I$ is positive definite, for any $\theta_2 > 0$, we can show that the dual of the problem (17) is given by

$$\begin{aligned} \max_{\mathbf{p}_i, \mathbf{u}_i} \quad & \frac{1}{2} \frac{\|\hat{G}^{-1/2}(\hat{\mathbf{p}}_2 - (1-\epsilon)\hat{\mathbf{p}}_1)\|^2}{(\|\hat{G}^{-1/2}\hat{\mathbf{p}}_1\| \|\hat{G}^{-1/2}\hat{\mathbf{p}}_2\|)^2 - (\hat{\mathbf{p}}_1^\top \hat{G}^{-1} \hat{\mathbf{p}}_2)^2} \\ \text{s.t.} \quad & \mathbf{p}_1 \in \mathbf{B}(\boldsymbol{\mu}_1, S_1, \kappa_3), \quad \mathbf{p}_2 \in \mathbf{B}(\boldsymbol{\mu}_3, S_3, \kappa_4), \end{aligned} \tag{A.40}$$

where $\hat{\mathbf{p}}_i = [\mathbf{p}_i; 1] \in \Re^{n+1}$, for $i = 1, 2$.

References

1. Agarwal S, Tomar D (2014) A feature selection based model for software defect prediction. Int J Adv SciTechnol 65:39–58
2. Angulo C, Parra X, Catal A (2003) K-SVCR: a support vector machine for multi-class classification. Neurocomputing 55:57–77

3. Bache K, Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
4. Bosch P, López J, Ramírez H, Robotham H (2013) Support vector machine under uncertainty: An application for hydroacoustic classification of fish-schools in Chile. Expert Syst Appl 40(10):4029–4034
5. Bottou L, Cortes C, Denker J, Drucker H, Guyon I, Jackel L, LeCun Y, Muller U, Sackinger E, Simard P, Vapnik V (1994) Comparison of classifier methods: a case study in handwritten digit recognition. Proc Int Conf Pattern Recog 2:77–82
6. Bredensteiner EJ, Bennett KP (1999) Multicategory classification by support vector machines. Comput Optim Appl 12:53–79
7. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2011) Dbsmote: density-based synthetic minority over-sampling technique. Appl Intell 36:1–21
8. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
9. Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2:265–292
10. Demšar J (2006) Statistical comparisons of classifiers over multiple data set. J Mach Learn Res 1–30
11. Dinkelbach W (1967) On nonlinear fractional programming. Manag Sci 13:492–498
12. Djuric N, Lan L, Vucetic S, Wang Z (2013) Budgetedsvm: a toolbox for scalable svm approximations. J Mach Learn Res 14:3813–3817
13. Friedman J (1996) Another approach to polychotomous classification. Tech. rep., Department of Statistics, Stanford University, http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z
14. Geng X, Zhan DC, Zhou ZH (2005) Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Trans Syst Man Cybern Part B: Cybern 35(6):1098–1107
15. Goldfarb D, Iyengar G (2003) Robust convex quadratically constrained programs. Math Program 97(3):495–515
16. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70
17. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
18. Kim YJ, Baik B, Cho S (2016) Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning. Expert Syst Appl 62:32–43
19. Kressel UG (1999) Advances in kernel methods. MIT Press, Cambridge, pp 255–268. USA, chap Pairwise classification and support vector machines
20. Lanckriet G, Ghaoui L, Bhattacharyya C, Jordan M (2003) A robust minimax approach to classification. J Mach Learn Res 3:555–582
21. Le Thi H, Pham Dinh T, Thiao M (2016) Efficient approaches for l2-l0 regularization and applications to feature selection in svm. Appl Intell 45(2):549–565
22. López J, Maldonado S (2016) Multi-class second-order cone programming support vector machines. Inform Sci 330:328–341
23. López J, Maldonado S, Carrasco M (2016) A novel multi-class svm model using second-order cone constraints. Appl Intell 44(2):457–469
24. Maldonado S, López J (2014) Imbalanced data classification using second-order cone programming support vector machines. Pattern Recogn 47:2070–2079
25. Maldonado S, López J, Carrasco M (2016) A second-order cone programming formulation for twin support vector machines. Appl Intell 45(2):265–276
26. Mangasarian OL (1994) Nonlinear programming. Classics in applied mathematics, society for industrial and applied mathematics
27. Mercer J (1909) Functions of positive and negative type, and their connection with the theory of integral equations. Philos Trans R Soc Lond 209:415–446
28. Nath S, Bhattacharyya C (2007) Maximum margin classifiers with specified false positive and false negative error rates. In: Proceedings of the SIAM international conference on data mining
29. Qi Z, Tian Y, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recogn 46(1):305–316
30. Sánchez-Morillo D, López-Gordo M, León A (2014) Novel multiclass classification for home-based diagnosis of sleep apnea hypopnea syndrome. Expert Syst Appl 41(4):1654–1662
31. Schölkopf B, Smola AJ (2002) Learning with kernels. MIT Press
32. Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
33. Sturm J (1999) Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim Methods Softw 11(12):625–653. Special issue on Interior Point Methods (CD supplement with software)
34. Tomar D, Agarwal S (2014) Feature selection based least square twin support vector machine for diagnosis of heart disease. Int J Bio-Sci Bio-Technol 6(2):69–82
35. Vapnik V (1998) Statistical learning theory. Wiley
36. Wang Z, Crammer K, Vucetic S (2010) Multi-class pegasos on a budget. In: Proceedings of the 27th international conference on machine learning (ICML-10). Omnipress, pp 1143–1150
37. Wang Z, Djuric N, Crammer K, Vucetic S (2011) Trading representability for scalability: adaptive multi-hyperplane machine for nonlinear classification. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 24–32
38. Wang Z, Crammer K, Vucetic S (2012) Breaking the curse of kernelization: budgeted stochastic gradient descent for large-scale SVM training. J Mach Learn Res 13:3103–3131
39. Weston J, Watkins C (1999) Multi-class support vector machines. In: Proceedings of the Seventh European symposium on artificial neural networks
40. Weston J, Elisseeff A, BakIr G, Sinz F (2005) The spider machine learning toolbox. http://www.kyb.tuebingen.mpg.de/bs/people/spider/
41. Xie J, Hone K, Xie W, Gao X, Shi Y, Liu X (2013) Extending twin support vector machine classifier for multi-category classification problems. Intell Data Anal 17(4):649–664
42. Xu Y, Guo R, Wang L (2013) A twin multi-class classification support vector machines. Cogn Comput 5:580–588
43. Yang HY, Wang XY, Niu PP, Liu YC (2014) Image denoising using nonsubsampled shearlet transform and twin support vector machines. Neural Netw 57:152–165
44. Yang Z, Shao Y, Zhang X (2013) Multiple birth support vector machine for multi-class classification. Neural Comput Appl 22:S153–S161
45. Zeng M, Yang Y, Zheng J, Cheng J (2015) Maximum margin classification based on flexible convex hulls. Neurocomputing 149(B):957–965
46. Zhong P, Fukushima M (2007) Second-order cone programming formulations for robust multiclass classification. Neural Comput 19:258–282

Julio López received his B.S. degree in Mathematics in 2000 from the University of Trujillo, Perú. He also received the M.S. degree in Sciences in 2003 from the University of Trujillo, Perú and the Ph.D. degree in Engineering Sciences, minor Mathematical Modelling in 2009 from the University of Chile. Currently, he is an Assistant Professor of Institute of Basic Sciences at the University Diego Portales, Santiago, Chile. His research interests include conic programming, convex analysis, algorithms and machine learning.

Sebastián Maldonado received his B.S. and M.S. degree from the University of Chile, in 2007, and his Ph.D. degree from the University of Chile, in 2011. He is currently Associate Professor at the Faculty of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile. His research interests include statistical learning, data mining and business analytics.

Miguel Carrasco received his B.S. degree in Mathematics in 2002 and the B.S. degree in Computing Sciences in 2005 from the University of Chile. He also received the Ph.D. degree in Engineering Sciences, minor Mathematical Modelling in 2007 from the University of Chile in collaboration with University of Montpellier II, France. Currently, he is full time professor at the Faculty of Engineering and Applied Sciences, Universidad de los Andes, Santiago, Chile. His research interests include convex analysis, proximal type algorithms, conic programming and topology optimization.